BHAVANA CK

1BM20CS403

CSE -4A

PROGRAM 4: STUDENT FACULTY DATABASE

Consider the following database for student enrollment for course :

STUDENT(snum : integer, sname : string, major: string, lvl : string, age: integer)

CLASS(cname: string, meets at: time, room: string, fid: integer)

ENROLLED(snum: integer, cname: string)

FACULTY(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)

Write the following queries in SQL. No duplicates should be printed in any of the answers.

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by

ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

iii. Find the names of all students who are enrolled in two classes that meet at the same time.

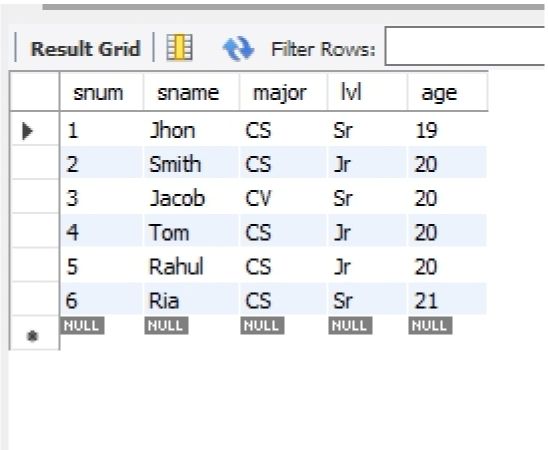
iv. Find the names of faculty members who teach in every room in which some class is taught.

v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

vi. Find the names of students who are not enrolled in any class.

vii. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

```
create database studentfaculty;
use studentfaculty;

create table student(
snum int not null,
sname varchar(20) not null,
major varchar(2) not null,
lvl varchar(2) not null,
age int not null,
primary key (snum)
);


create table faculty(
fid int not null,
fname varchar(20) not null,
deptid int not null,
primary key(fid)
);


create table class(
cname varchar(20) not null,
meetsat datetime not null,
room varchar(4) not null,
fid int not null,
primary key (cname),
foreign key(fid)references faculty(fid)
);


create table enrolled(
snum int not null,
cname varchar(20) not null,
primary key(snum,cname),
foreign key(snum)references student(snum),
foreign key(cname) references class(cname )
);


insert into student values (1,"Jhon","CS","Sr",19) ,
(2,"Smith","CS","Jr",20),
(3,"Jacob","CV","Sr",20),
(4,"Tom","CS","Jr",20),
(5,"Rahul","CS","Jr",20),
(6,"Ria","CS","Sr",21);
```



snum	sname	major	lvl	age
1	Jhon	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Ria	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

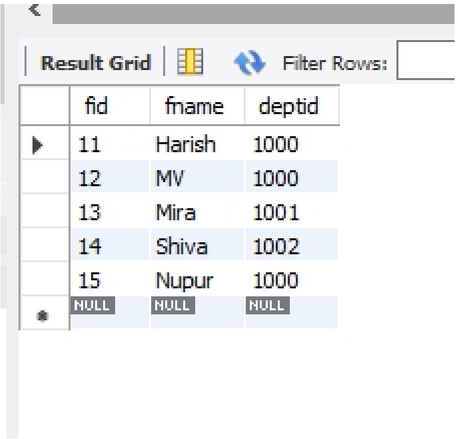
insert into faculty values(11,"Harish",1000),

(12,"MV",1000),

(13,"Mira",1001),

(14,"Shiva",1002),

(15,"Nupur",1000);



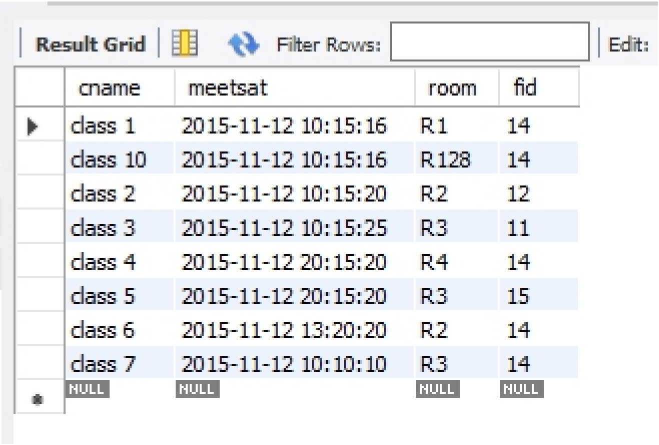
fid	fname	deptid
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
NULL	NULL	NULL

insert into class values("class 1","2015-11-12 10:15:16","R1",14),

("class 10","2015-11-12 10:15:16","R128",14),

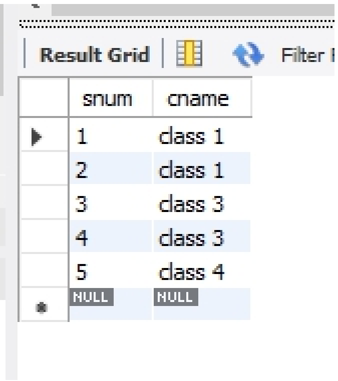
("class 2","2015-11-12 10:15:20","R2",12),

```
("class 3","2015-11-12 10:15:25","R3",11),

("class 4","2015-11-12 20:15:20","R4",14),

("class 5","2015-11-12 20:15:20","R3",15),

("class 6","2015-11-12 13:20:20","R2",14),

("class 7","2015-11-12 10:10:10","R3",14);
```



cname	meetsat	room	fid
class 1	2015-11-12 10:15:16	R1	14
class 10	2015-11-12 10:15:16	R128	14
class 2	2015-11-12 10:15:20	R2	12
class 3	2015-11-12 10:15:25	R3	11
class 4	2015-11-12 20:15:20	R4	14
class 5	2015-11-12 20:15:20	R3	15
class 6	2015-11-12 13:20:20	R2	14
class 7	2015-11-12 10:10:10	R3	14
NULL	NULL	NULL	NULL

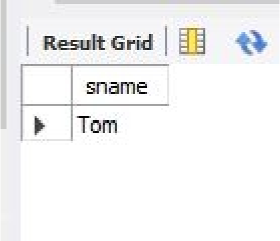
```
insert into enrolled values(1,"class 1"),

(2,"class 1"),

(3,"class 3"),

(4,"class 3"),

(5,"class 4");
```



snum	cname
1	class 1
2	class 1
3	class 3
4	class 3
5	class 4
NULL	NULL

------ Find the names of all Juniors (level = JR) who are enrolled in a class taught by

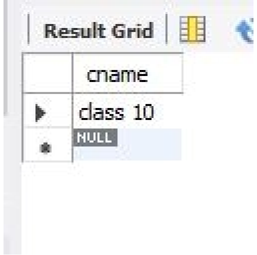
```
select s.sname from student s,enrolled e,class c

where s.snum=e.snum and c.cname = e.cname and c.fid =(select fid from faculty

where fname ="Harish")and s.lvl="Jr";
```



```
sname
Tom
```

------ Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

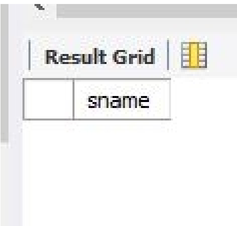
```
select c.cname from class c

where c.room = "R128"

or c.cname in(select e.cname from enrolled e group by e.cname having count(*)>=5);
```



```
cname
class 10
NULL
```

------ Find the names of all students who are enrolled in two classes that meet at the same time.

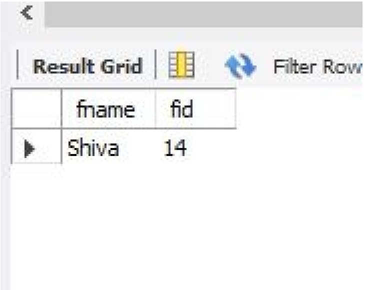
```
select distinct s.sname from student s

 where s.snum in(select e1.snum from enrolled e1,enrolled e2,class c1,class c2

 where e1.snum=e2.snum and e1.cname<>e2.cname and e1.cname = c1.cname

 and e2.cname=c2.cname and c1.meetsat=c2.meetsat);
```



sname

------ Find the names of faculty members who teach in every room in which some class is taught.

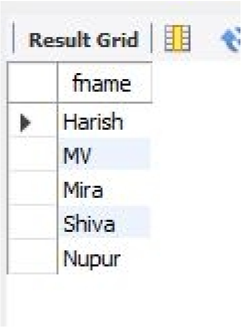
```
select f.fname,c.fid from faculty f,class c
 where f.fid = c.fid
 group by c.fid
 having count(c.fid)=(select count(distinct room) from class);
```



fname	fid
Shiva	14

------ Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

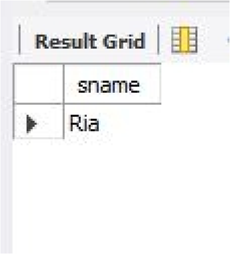
```
select distinct fname from faculty f
 where 5>(select count(e.snum)from enrolled e,class c
 where c.cname = e.cname and c.fid = f.fid);
```



fname
Harish
MV
Mira
Shiva
Nupur

------ Find the names of students who are not enrolled in any class.

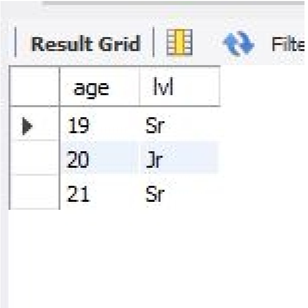
```
select s.sname from student s

where snum not in(select snum from enrolled);
```



	sname
▶	Ria

------ For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

```
select s.age,s.lvl from student s

group by s.age having s.lvl in(select s1.lvl from student s1

where s1.age = s.age group by s1.age having count(*)>=all(select s2.lvl from student s2

where s2.age = s1.age group by s2.age));
```



	age	lvl
▶	19	Sr
	20	Jr
	21	Sr